SYSTEM AND METHOD FOR SELECTIVE MEMORY MODULE POWER MANAGEMENT

TECHNICAL FIELD

This invention relates to computer memory systems. More particularly, the present invention relates to enhancing power management and reducing power consumption in a computer memory system.

BACKGROUND OF THE INVENTION

Most computers and other digital systems have a system memory which often consists of dynamic random access memory ("DRAM") devices. DRAM devices are fairly inexpensive because a DRAM memory cell needs relatively few components to store a data bit as compared with other types of memory cells. Thus, a large system memory can be implemented using DRAM devices for a relatively low cost.

Commonly, DRAM devices are arranged on memory modules, such as single in-line memory modules ("SIMMs") and dual in-line memory modules ("DIMMs"). A representative module is shown in Figure 1. The module 100 features a number of DRAM devices 104 mounted on an insulative substrate 108 through which the DRAM devices 104 are operably coupled through communications lines 110 such as conductive traces or other similar signal carrying devices to a memory hub 112. The module 100 interfaces with a system (not shown) through a series of conductive terminals 116 or other means through which control, data, and address information is communicated between the system and the module 100. A typical memory module 100 may support a number of DRAM devices 104 which supports an array of single-bit storage devices. A number of these DRAM devices 104 are arrayed in a parallel fashion such that, upon the module 100 receiving a specified address, the memory hub 112 will cause a data bit stored at the same address in each of the array of memory devices 104 to be retrieved to effectively retrieve a full data word. For example, if the memory module 100 features eight DRAM devices 104, each address applied to the module 100, the memory hub 112 will cause an eight-bit byte to be retrieved from the DRAM devices 104.

The proliferation of this modular design has a number of advantages, ranging from the ability to provide a large memory capacity in a relatively small package to greatly simplifying the installation process as compared to the painstaking process of installing individual memory chips. Beyond these more obvious advantages of modular design, however, is the additional functionality which is made possible by the use of the memory hub 112 (Figure 1). To name one example, the memory hub 112 can include one or more registers, allowing address, data, and/or control information to be latched. The latching of this information allows for synchronous operations using this information without concern for data transiency problems such as race, skew, or synchronization problems which might result if the module had to be perfectly in synchronization with the system bus in receiving and outputting data. In addition, computer systems employing this architecture can have a higher bandwidth because a processor can access one memory device while another memory device is responding to a prior memory access. For example, the processor can output write data to one of the memory devices in the system while another memory device in the system is preparing to provide read data to the processor. Continually, new techniques are being developed to exploit the control permitted by the presence of the memory hub 112 central control logic on these memory modules 100.

Returning to the DRAM devices themselves, while DRAM devices do provide a relatively inexpensive way to provide a large system memory, DRAM devices suffer from the disadvantage that their memory cells must be continually refreshed. Refreshing memory cells consumes an appreciable quantity of power. Because of this drain of power, an important topic in DRAM design is how to reduce the power consumed in refreshing DRAM cells.

Once such technique for reducing power consumption is the implementation of a self-refresh cycle. Figure 2 depicts a block diagram of a conventional DRAM device 200 enabled to use self-refresh. The DRAM device 200 is accessed through the address lines 210, the data lines 212, and a number of control lines 220-232. These control lines include CKE (clock enable) 220, CK* (clock signal – low) 222, CK (clock signal) 224, CS* (chip select – low enable) 226, WE* (write select – low enable) 228, CAS* (column address strobe – low enable) 230, and RAS* (row address strobe – low enable) 230. The address lines 210, data lines 212, and control lines 220-232, enable the system to read and write data to the actual memory banks 250, as well as control the refreshing of the DRAM device 200. The control logic 260 controls the read, write, and refresh operations of the DRAM device 200. The control logic 260 directs the operations of the DRAM device 200 as a function of the signals received at the control lines 220-232.

A DRAM device 200 typically is refreshed using an auto-refresh cycle, which is triggered by the system and operates synchronously with the system clock. More specifically, with the CKE 220 and WE* 228 control lines driven high, and the CS* 226, RAS* 230 and CAS* 232 control lines driven low, the rising edge of the next clock signal initiates an auto-refresh of the next row of the memory banks 250. Once the system initiates an auto-refresh cycle, the refresh counter 270 is incremented by one, and the row of the memory banks 250 corresponding to the updated count stored in the refresh counter 270 is refreshed. The refresh counter 270 maintains its count to track what row is next to be refreshed when the next auto-refresh cycle is initiated. This process repeats continuously. In a typical DRAM, having 4,096 rows and a maximum refresh interval of 64 milliseconds in its operational mode, a command to refresh one row would have to be issued approximately every 15 to 16 microseconds.

Although the auto-refresh process is a relatively simple one, auto-refresh requires that hundreds or thousands of times per second, thousands of control logic and access transistors within the devices depicted in Figure 2 and described in the foregoing description must be energized to refresh the array, consuming power. In addition, resistance of the conductors through the memory array to address each and every transistor in each and every row consumes even more power. Still more power is consumed by transistors used in the sense amplifiers which read and refresh the memory cells in respective columns. Moreover, power is needed to actually charge each of the thousands of capacitors storing data bits in the array.

Implementation of a self-refresh cycle saves some of the power consumed as compared with auto-refresh. Initiation of a self-refresh cycle places a DRAM device 200 in a continual, indefinite refresh cycle to preserve the data stored in the DRAM device 200. A self-refresh command typically is issued during a period when useful read and write requests will not be forthcoming, for example, when a user has placed the computing system into a sleep or standby mode. A self-refresh command is triggered by driving the CS* 226, RAS* 230 and the CAS* 232 control lines low, driving the WE* 228 control line high, and, this time, driving the CKE 220 control line low. This command causes the self-refresh control logic 280 to periodically and repeatedly refresh every one of its rows, and also places all data, address, and control lines into a "don't care" state, with the exception of the CKE 220 control line. Driving the CKE 220 control line high ends the self-refresh state, removing the other control lines out of the "don't care" state.

During a self-refresh cycle, with most of the control lines in a don't care state, devices in the DRAM device 200 will not be switching to decode memory addresses and perform read or write commands, thus current and voltage fluctuations in the DRAM device 200 are reduced. This relatively stable condition tends to ameliorate electrical and thermal effects which contribute to current leakage from the capacitors of the memory cells. As a result, while the memory cells still need to be refreshed to preserve the integrity of the data stored therein, the memory cells do not need to be refreshed as frequently as during an operational state. During self-refresh, the contents of the memory cells can be preserved by refreshing a row less frequently than required during normal operation. In self-refresh state, for example, the rows might not need to be refreshed for a period up to twice as long, or perhaps slightly longer, than is permitted during an operational state.

While self-refresh can save an appreciable amount of power, self-refresh traditionally is implemented on a system-wide basis, often along with other power-saving techniques: For example, when a computer is placed in a standby mode, virtually every device in the computer enters a standby mode, i.e., the display is shut down, the hard disk is stopped, the memory is placed in a self-refresh state, and other systems are similarly put to "sleep."

Operating systems, such as Windows 2000® do allow for more advanced power management options, and a user can select an interval of disuse after which the hard disk, the display, and the entire system will power down. In addition, some operating systems or utilities provide for additional power management choices allowing a user to choose operating parameters ranging between maximum performance at one extreme and maximum power savings at another extreme, or some intermediate compromise choice to suit the user's preferences. Still, while all these options save power, the only means to avoid wasting power in system memory remains an all or nothing, standby or not proposition.

What is needed is a way to save power which might be wasted in system memory. It is to this end that the present invention is directed.

SUMMARY OF THE INVENTION

A memory module is equipped with means to monitor utilization of the memory module. Through these devices, system utilization of the memory module can be monitored by tracking actual system usage, such in the form of read and write commands issued to the memory module, or by measuring temperature changes that indicate a nominal level of read and write activity beyond continual refresh activity. According to one aspect of the invention, control logic on the memory module directs the memory module into a power saving mode after determining, responsive to current activity levels, that the module need not remain immediately ready to process memory commands. In accordance with another aspect of the invention, the control logic could throttle activity of the memory module to reduce the responsiveness of the memory module in the face of receiving more than a desired number of system commands per unit time and/or measured temperature levels or changes. In such a mode, the memory module would not be rendered dormant to system operations as in the previously described aspect of the invention, but instead would merely limit memory module usage and allow the memory module to process only a predetermined number of system commands or remain at or below a certain operating temperature. For example, the control logic would cause a number of idle states to be observed to maintain memory module power consumption below a certain level.

According to another aspect of the invention, data packets summarizing the memory module's activity level are transmitted on the memory bus via the memory hub. The memory module activity level packet could be received by a memory controller or by a master memory hub disposed on another memory module. Selectively directing memory modules into a reduced power state can thereby be managed centrally by the system controller, the memory controller, or a master memory module equipped with a master module power management controller. The system controller or master module power management controller may also communicate power control data packets to other memory modules via the system bus and the other memory modules' memory hubs to direct those modules into reduced power states. The system controller or master module power management controller could direct the memory modules into a power saving mode such as a self-refresh mode, could throttle memory module activity to reduce responsiveness and reduce power consumption, or use another reduced power mode.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a plan view of a conventional memory module.

Figure 2 is a block diagram of a conventional memory device equipped with self-refresh circuitry.

Figure 3 is a plan view of a memory module equipped with power saving facilities of an embodiment of the present invention.

Figure 4 is a flowchart showing the power saving operations of a memory module equipped with an embodiment of the present invention.

Figure 5 is a plan view of a plurality of memory modules equipped with activity monitoring capabilities and communicating activity packets on the memory bus to a master power controller of another embodiment of the present invention.

Figure 6 is a block diagram of a computer system employing an embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

Figure 3 shows a memory module 300 equipped with activity monitoring and power saving capabilities employing a first embodiment of the present invention. The memory module 300 comprises a plurality of memory devices 104 mounted on a substrate 108 through which the DRAM devices 104 are operably coupled to a memory hub 312 through communications lines 110 such as conductive traces or other similar signal carrying devices. The memory module shown in Figure 3 comprises most of the same components used in the memory module shown in Figure 1 thus, in the interest of brevity, these components have been provided with the same reference numerals, and an explanation of their functions and operations will not be repeated.

The memory module 300 shown in Figure 3 comprises three additional devices not included in the conventional memory module of Figure 1. The memory module 300 includes an activity monitor 350, a power management controller 360, and a temperature sensor 370, the last being connected to the memory devices 104 via a network of connections 380. Generally, the power management controller 360 monitors signals received from the activity monitor 350 and the temperature sensor 370 to determine whether the memory module 300 is active. If the memory module 300 is active, it is maintained at fully operational status. However, if the memory module 300 is not active, and the power management controller 360 can direct the memory module 300 to assume a reduced power consumption state. The activity monitor 350 actually tracks memory commands to the memory module 300, such as read and write requests to that module, to directly gauge whether the system is using the memory module. The temperature sensor 370 tracks the temperature of the memory devices 104 to indirectly measure whether the system is using the memory devices. As is known in the art, memory devices 104 actually being used consume more power and radiate more heat than memory devices 104 not being actively used, because additional circuitry is required to respond to memory commands than to merely continually refreshing the memory devices' own memory cells.

The power management controller 360, acting on input from the activity monitor 350 or the temperature sensor 370, can direct the memory module 300 into a reduced power mode when the memory module is inactive. For example, the memory module 300 might be inactive if it represents a portion of memory configured to be at the upper end of the system memory, and the user is not running applications requiring enough memory to load programs or data into that portion of memory. Alternatively, the memory devices 104 on the memory module 300 might have been loaded with programs and data the user is not actively using. For example, the memory devices 104 on the memory module 300 might have been loaded with a word processing document the user opened and has left idle in an open window, while the user works with a program loaded into memory devices on other memory modules (not shown). In addition, the user may have stopped using the system altogether for a few moments, resulting in none of the contents stored in the memory devices 104 and memory modules actively being used for a time. Such examples of lack of activity may signal that these memory devices 104 could be directed into a power saving state. The activity monitor 350 might count memory commands directed to the memory module 300, and after counting a predetermined number of clock cycles corresponding to a preselected time interval without a memory command, the activity monitor 350 could signal the power management controller 360 that the memory module 300 could assume a lower power consumption state.

In Figure 3, the activity monitor 350 and the power management controller 360 are shown as being a part of the memory hub 312. Because memory commands would be received by the memory hub 312, it is a logical choice to incorporate the device monitoring system activity, the activity monitor 350, within the memory hub 312 itself. Similarly, because the memory hub 312 is in communication with the memory devices 104, it is a logical choice to include the power management controller 360 in the memory hub as well. However, the activity monitor 350 and/or the power management controller 360 can alternatively be located elsewhere in the memory module 300. The temperature sensor 370 is shown in Figure 3 as being external to the memory hub 312 and connected to each of the memory devices 104 through the network of connectors 380. This is one of a number of possible designs, as will be further described in connection with describing the operation of the temperature sensor 370.

In one embodiment, the activity monitor 350 (Figure 3) might be a counter to track the number of clock cycles since the last memory request from the system. After a sufficiently large predetermined number of clock cycles has passed without a memory command, an overflow signal on the counter might signal to the power management controller 360 (Figure 3) that this threshold has been reached. Reaching this threshold count could be taken as an indication that the system is not using the memory module 300 or, at least, not presently using any contents of the memory module.

In addition to directly monitoring memory commands, a memory module 300 equipped with this embodiment of the present invention also can determine system activity somewhat less directly by measuring the temperature of the memory devices 104. As is well understood in the art, semiconductor devices such as memory devices consume power, some of which is lost to waste heat, with the more activity taking place in the device, the greater the amount of heat generated. As is known in the art, when a device is actively being used, more gates and other circuits in the device will be switching; the more circuits that are switching, the more power the device draws, and more heat is generated. To give an example, in a memory device 104, refreshing the memory array in a system-directed, ordinary auto-refresh mode consumes less power than the same semiconductor device actually processing memory commands, and therefore generates less heat.

The temperature sensor 370 can be deployed in a number of different ways. As shown in Figure 3, the temperature sensor 370 is connected to each of the memory devices 104 through a network of communicative connections. The memory devices 104 can each be equipped with a temperature sensor device which communicates an electrical signal to the temperature sensor 370, which can discern an average temperature level across the array memory devices 104. Alternatively, the temperature sensor 370 could be connected to one memory device 104 or a number of representative memory devices 104, taking the operating temperature of that sampling of memory devices 104 as being indicative of the operating temperature of each of the memory devices 104. In addition, the temperature sensor 370 could measure the temperature of the substrate 108, which would change in response to the heat generated by the memory devices 104 as their activity level varies.

The temperature sensor 370 will compare the measured temperature to a predetermined threshold temperature. This temperature can be specified as an absolute value, as an absolute value relative to an ambient system temperature which might be measured by or communicated to the temperature sensor 370, or as a differential measured from an operating temperature reached by the memory module 300 once it has become fully operational. Alternatively, the temperature sensor 370 could be programmed to respond to a combination of factors, for example, when the temperature falls below a predetermined threshold and when that temperature represents a predetermined differential from a previously measured operating temperature. Once the temperature sensor 370 detects that the threshold or thresholds have been reached, the temperature sensor 370 might signal the power management controller 360 that the temperature level indicates the memory module 300 has not been actively used, and could assume a reduced power state.

As mentioned above, the power management controller 360 receives signals from the activity monitor 350 and the temperature sensor 370 and, responsive to those signals, determines when the memory module might be directed to a reduced power state and restored to fully operational status. Figure 4 flowcharts the operation of the invention the power management controller 360 (Figure 3), the activity monitor 350, and the temperature sensor 370. Starting with the memory module 300 (Figure 3) at operational status and consuming a full quantity of power from a system start or other fully operational status at 404, the activity monitor 350 (Figure 3) is engaged to monitor memory commands issued to the memory module 300 (Figure 3) at 408 (Figure 4) as previously described. The temperature sensor 370 (Figure 3) also is engaged to monitor the operating temperature of the memory devices (Figure 3) at 412 (Figure 4) on the module as previously described.

From the time these devices are engaged, the power management controller 360 (Figure 3) continuously monitors the signals received from these devices. If the number of memory commands received continues to indicate that the memory module 300 (Figure 3) is in regular, active use at 416 (Figure 4), and the operating temperature of the memory devices 104 (Figure 3) continues to indicate the same at 424 (Figure 4), the power management controller 360 (Figure 3) maintains the memory module at full operational status and power. Nonetheless, as shown in Figure 4, the memory management controller 360 continues to monitor the status of these signals.

On the other hand, if the activity monitor 350 (Figure 3) signals that no memory commands have been received for a period reaching an idle threshold at 416, or the temperature level indicates that the memory module 300 (Figure 3) has not been actively used at 424 (Figure 4), the power management controller 360 (Figure 3) may direct the memory module 300 into a reduced power mode at 420 (Figure 4). As previously described, this power reduction state might be a self-refresh mode during which the memory devices 104 (Figure 3) are effectively isolated from the system and thus can be refreshed at a reduced rate, saving power. The memory module 300 can continue in this reduced power state until a memory command is received at 428 (Figure 4) as detected by the activity monitor 350 (Figure 3). Upon receiving such a memory command, the memory module 300 can resume its fully operational power status at 404 (Figure 4), resetting the activity monitor 350 (Figure 3) and/or the temperature sensor 370 to await the next time when the memory module 300 becomes idle and can assume a reduced power mode.

Although power saving techniques for memory systems such as self-refresh are currently known and used in computer systems, one of the advantages of embodiments of the present invention is that such techniques can be applied selectively. Conventionally, power-saving techniques are implemented across the entire system when a system user manually directs the system into a standby mode, or when the system automatically transitions into a standby mode after a predetermined period of inactivity. Embodiments of the present invention, however, allow for reaping these power savings while a system is operating. As a result, embodiments of the present invention can extend the actual operating time of electronic aids employing such memory devices.

It should be understood that use of the self-refresh mode is not the only possible way that embodiments of the present invention can be used to save power in memory systems. To name one example, the power management controller 360 (Figure 3), through its associated activity monitor 350, might detect that no data has been loaded into the memory devices 104 of the memory module 300. If the memory module 300 is completely idle, as might be the case when the user is not running sufficient applications to fully utilize the system memory, the memory devices 104 could be powered off, along with the temperature sensor 370 and other devices. As long as the memory hub 312 and the power management controller 360 in the present example were left powered on to detect a memory command directed to the memory module 300 and so that the memory devices 104 and other dormant devices can be powered on again, further power can be saved. Similarly, a memory module 300 whose memory devices 104 store contents that have been long dormant could dump their contents to disk storage or other storage, and power down the devices. Upon receiving a memory command, the contents could be restored from disk to memory, allowing the user to continue the application from where she last was. Windows 2000® incorporates a "hibernate" mode that allows the entire system to shut down in this manner, allowing for a quick restart. However, as with other power saving facilities currently in use, the "hibernate" mode is an all-or-nothing, system wide shut down, and not applied selectively to some or all of the memory devices, as could be using embodiments of the present invention.

Alternatively, rather than direct the memory module 300 into an inactive state, the power management controller 360 can "throttle" the activity of the memory module 300 to system commands to limit power consumption. Instead of directing the memory module 300 into a nonfunctional state, such as a self-refresh state, throttling activity of the memory module 300 will reduce the responsiveness of the memory module 300 to keep its power consumption at or below a desired level. The power management controller 360 may be directed to restrict the number of system commands processed by the memory module 300 per unit time, mandating a certain number of idle intervals pass after one or a number of system commands have been processed per unit time. In one embodiment, the power management controller 360 may be programmed to always respond to a first system command or a first number of system commands, then insert a requisite number of idle intervals to contain power consumption. Alternatively, the power management controller 360 might evaluate power consumption by monitoring device temperatures, correlating a certain temperature level or change of temperature with exceeding a desired level of power consumption. As in the case of the power management controller 360 monitoring system requests, after the power management controller 360 measures a certain temperature level or change, the power management controller 360 can mandate a number of idle states, during which power consumption and, therefore, device temperature will decrease. Throttling the activity of the memory module 300 in this way, its power consumption can be reduced without actually rendering the memory module 300 at least temporarily inactive, as in the case of directing the memory module 300 into self-refresh mode.

Another embodiment of the present invention is shown in Figure 5. Figure 5 shows a network of two memory modules 504 and 508 coupled with a memory bus 512 to a system controller or memory controller 516. The memory modules 504 and 508 are nearly identical to the memory module 300 shown in Figure 3, each having one difference. Memory module 504, positioned closest to the system/memory controller 516 is installed as the primary, low address memory module, and its memory hub 528 includes a primary power management controller 520. The memory hub 532 of memory module 508 includes a secondary power management controller 524. The primary power management controller 520 and the secondary power management controller 524 operate in a master/slave arrangement. Information about the activity in the secondary memory module 508 is relayed through the memory hub 532 over the system bus 512 to the primary memory module 504 and the primary power management controller 520. Similar to the operations of the memory module 300 of Figure 3, the primary power management controller 520 also receives information about its own activity level.

Responsive to information received about its own activity level, the activity level of the secondary memory module 508, and any other memory modules (not shown) associated with the system, the primary power management controller 520 determines whether its own devices, those on the secondary memory module 508, or any other memory modules (not shown) should be directed to a reduced power state. As will be appreciated, these control decisions are made by the primary power management controller 520 just as they were made by the power management controller 360 of the memory module 300 of Figure 3, which, for example, were based on activity level as reflected in actual system usage of these memory modules or by temperature levels reflecting the level of device activity. The primary power management controller 520 directs devices on the secondary memory module 508 by transmitting a control packet through its memory hub 528 via the system bus 512 to the secondary power management controller 524. On receiving a reduced power directive, the secondary power management controller 524 directs devices on the memory module to a reduced power state, whether that be a self-refresh state, a powered off state, a throttling or reduced response mode as previously described, or another reduced power state.

It will be appreciated that, in such a centralized control system, all the same power saving techniques could be employed. Memory devices 104 could be directed into a reduced power mode. Alternatively, the memory devices 104 and other devices could be powered off entirely if unused, or after having long dormant contents archived, both as previously described. As long as devices on the secondary memory module 508 remain active such that the secondary memory module 508 can be reactivated when memory commands to the secondary memory module are received, power can be saved in avoiding refreshing empty or long-unused and archived data.

A computer system 600 using the memory modules 300 of Figure 3 or 504 and 508 of Figure 5 according to examples of the present invention are shown in Figure 6. The computer system 600 includes a processor 614 for performing various computing functions, such as executing specific software to perform specific calculations or tasks. The processor 614 includes a processor bus 618 that normally includes an address bus, a control bus, and a data bus. The computer system 600 includes a system controller 620 that is coupled to the processor bus 618. The system controller 620 also includes a memory controller 624, which is, in turn, coupled to memory modules 628a, 628b, 628c, and 628d through a system bus 632. It will be appreciated that that the controller 624 may be external to the system controller 620 and coupled to it or some other component in the computer system 600, such as the processor 614.

In addition, the computer system 600 includes one or more input devices 636, such as a keyboard or a mouse, coupled to the processor 614 through the system controller 620 to allow an operator to interface with the computer system 600. Typically, the computer system 600 also includes one or more output devices 640 coupled to the processor 614 through the system controller 620, such output devices typically being a printer or a video terminal. One or more data storage devices 644 are also typically coupled to the processor 614 through the system controller 620 to allow the processor 614 to store data or retrieve data from internal or external storage media (not shown). Examples of typical storage devices 640 include hard and floppy disk drives, removable large capacity disk drives, tape cartridge drives, removable flash EEPROM storage devices, and compact disc (CD) read-only, writeable, and rewriteable drives. The processor 614 is also typically coupled to cache memory 648, which is usually static random access memory ("SRAM").

From the foregoing it will be appreciated that, although specific embodiments of the invention have been described herein for purposes of illustration, various modifications may be made without deviating from the spirit and scope of the invention. Accordingly, the invention is not limited except as by the appended claims.